

ECE4873 Project Summary

Project Title	Dia-Bot, Installation Diagnostic Robot: 004-Dia-Bot (Diagnostic Robot) Platform
Team Members (names and majors)	Catherine "Grace" Kasper, CompE
	Connor Truono, CompE
	Andrew Galant, ME
	Jason Piotter, ME
	Hunter Present, ME
	Douglas Walker, ME
Course & Section Primary Advisor	ECE 4723 A/L01 Jianxin Jiao (ME), Vijay Madisetti (ECE)
Semester	2021/Fall Course: ECE 4723
Web Site URL	https://ecesenior design2021fall.ece.gatech.edu/sd21f09/
Project Abstract (250-300 words)	<p>Our team's sponsor, Vanderlande, is an industry-leading producer of logistic process automation with shuttle and conveyor systems. When installing such large transport systems, there are many potential issues which may be found. Therefore, manual verification of their structure and components can take a long time.</p> <p>Because of this issue, Vanderlande desires a tool to help verify their shuttle and conveyor systems during installation - namely, a diagnostic robot, or "Dia-Bot". This robot must be able to traverse the systems under its own power and provide necessary data to assist in quality assurance, ultimately reducing verification time.</p> <p>For system requirements, the Dia-Bot must be conveyable across all traditional conveyor surfaces (belts, rollers, curves, diverts) and able to traverse normal transitions (inclines, declines, merges, diverges). Additionally, it must be robust to withstand all system vibrations during its own movement and when being passively conveyed. While moving through the system, the bot must wirelessly transmit data to Vanderlande operators, most importantly a camera feed with 360-degree rotation and vertical tilting. Other sensors and algorithms should measure system data, such as vibration, sound, and temperature.</p> <p>The multidisciplinary team took a full approach to solving the mechanical, electrical, and software needs of creating a Dia-Bot prototype for proper concept validation for Vanderlande. The final design includes a vented metal main body frame to house and protect electrical components, continuous treads with a suspension system for navigating potentially tough conveyor terrain, and a full user interface with data processing abilities interacting with embedded components.</p> <p>Vanderlande operators can wirelessly connect to, control, and receive real-time images and data from the Dia-Bot. These movement, visuals, and data processing features allow operators to quickly identify potential installation problems during system verification, ultimately reducing required production time.</p>

Project Title	Dia-Bot, Installation Diagnostic Robot: 004-Dia-Bot (Diagnostic Robot) Platform
List codes and standards that significantly affect your project. Briefly describe how they influenced your design.	<p>NIOSH Lifting Equation: Factors: Frequency, Asymmetry, Coupling, Distance, Vertical, Horizontal</p> <ul style="list-style-type: none"> - No OSHA limit on object lifting weight - NIOSH equation predicts injury risk <p>There are no hard requirements on weight of objects in the workplace, but depending on how and how often objects are lifted, there may be more injury risk. With this and Vanderlande's size and weight requirements in mind, the mechanical team aimed to keep the bot's weight down while still being sturdy.</p>
List at least two significant realistic design constraints that applied to your project. Briefly describe how they affected your design.	<ol style="list-style-type: none"> 1. Wireless connectivity As the robot will be navigating through large conveyor systems away from operators, proper wireless connection must be established and all relevant data for images and control must be transmitted. This affected our choice of microprocessor, going with a Raspberry Pi 4 for its on-board Wi-Fi module and simple connectivity 2. Responsive software interface Many requirements needed to be represented in software, including data collection, processing, video streaming, and motor control. Due to the heavy load on an embedded processor, the software architecture had to be engineered with complex forms of multiprocessing and multithreading within processes. This architecture required much more effort to achieve in the Python environment but the constraints on interface responsiveness, in addition to other software requirements, all running on one processor to reduce electrical power used (another constraint), affected the programming process.
Briefly explain two significant trade-offs considered in your design, including options considered and the solution chosen.	<ol style="list-style-type: none"> 1. Web server vs. RealVNC: Exposing a user interface For wireless real-time communication with the bot, we could have set up an HTTP server on the Pi to send individual commands via a web browser. This was ultimately more complicated and did not permit live data and video updates as easily as using the RealVNC Viewer and Server, allowing direct control of the Pi and on-board data storage 2. Treads vs Wheels: Ability to traverse rough conveyor terrain While wheels have many advantages over continuous treads, such as speed and efficiency, the choice was ultimately made to run treads with a suspension system due to the requirement for the bot to properly traverse conveyor surfaces and normal transitions, making the bot as flexible as possible

<p>Briefly describe the computing aspects of your projects, specifically identifying hardware-software tradeoffs, interfaces, and/or interactions.</p> <p><i>Complete if applicable; required if team includes CmpE majors.</i></p>	<p>A key goal of the Dia-Bot software was to create a GUI which was done in Python3. Tkinter (“Tk interface”) was used to create a GUI, implementing buttons, sliders, and text input for the user, and interfacing with matplotlib for live graphing and real-time data updates.</p> <p>Sensors, DC motors, and camera servos controlled via GPIO pins from the Python software. Requires I2C (accelerometer) and SPI (ADC, since no analog pins exist on the Pi) interfaces with external libraries.</p> <p>Camera feed and image capturing completed with the Picamera software stack. Since this has been deprecated on the newest Raspbian OS version, an older version had to be used until Python bindings are made for the replacement library. This camera preview, while controlled by the software, is overlaid on top of the TK GUI rather than interfaced into it.</p> <p>One of the main challenges with the software was the multiprocessing of Python. The Multiprocessing Python library used to ensure responsive GUI and real-time data on the 4-core Broadcom processor. Uses one main process for TK, and for each data type, one to collect data from GPIO, and another to process data and collect relevant metrics. This software architecture was easily the toughest computing aspect, but worthwhile and necessary for proper speed via parallel processing.</p>
---	--